

Introdução ao Linux:

Aula III: Gerenciamento de processos, acesso remoto e programas básicos (editores de texto, visualizadores de PDF)

Prof. Dr. Marcelo Bianchi

*Israel Dragone, Jamison Assunção
Leonardo Fabricius e Rafael Monteiro*

24 de julho de 2018



Gerenciamento de processos

O que é um processo?

Um processo é um programa em execução!

Algumas das características comuns em todos os processos são:

- ▶ número de identificação: **pid**
- ▶ identificação do processo “pai”(parent): **ppid**
- ▶ identificação de usuário e grupo
- ▶ endereço na memória, variáveis de ambiente, ...

Como podemos monitorar e interagir com os processos rodando na máquina?

Alguns comandos básicos de gerenciamento de processos são:
ps, top, kill, killall, pkill, CTRL-Z

Gerenciamento de processos

O comando **ps**

Um dos programas fundamentais no monitoramento de processos. O programa exibe uma *fotografia* dos processos rodando no momento de sua execução.

```
$ ps
```

```
PID TTY TIME CMD
24424 pts/0 00:00:00 bash
24468 pts/0 00:00:00 ps
```

```
$ ps -l
```

```
F S UID PID PPID C PRI NI ADDR SZ WCHAN TTY TIME CMD
0 S 1000 16258 13222 0 80 0 - 5329 - pts/1 00:00:00 bash
0 R 1000 16634 16258 0 80 0 - 7467 - pts/1 00:00:00 ps
```

Gerenciamento de processos

Identificação dos campos do comando `ps`

USER: nome do dono do processo

UID: número do dono do processo

PID: identificação do processo

PPID: identificação do processo pai (parent)

%CPU: porcentagem de cpu

%MEM: porcentagem de memória

VSZ: tamanho virtual

RSS: quantidade de memória usada

TTY: identificador do terminal

START: hora de início

TIME: tempo de atividade

COMMAND: nome do comando que executa o processo

PRI: valor da prioridade

NI: valor preciso da prioridade

WCHAN: função do kernel onde o processo está no modo suspenso

Gerenciamento de processos

O campo **STAT**

R: executando

D: esperando o disco

S: suspenso

T: interrompido

Z: zumbi

W: processo com página em disco

<: prioridade maior do que o comum

N: prioridade menor que o comum

L: alguns recursos bloqueados pelo kernel

Gerenciamento de processos

Algumas opções de visualização utilizando `ps`

- ▶ **ps a**: exibe processos criados por todos os usuários do sistema
- ▶ **ps u**: exibe o nome do usuário que iniciou o processo e a hora de inicialização
- ▶ **ps x**: exibe processos que não são controlados pelo terminal
- ▶ **ps m**: exibe a memória ocupada por cada processo

Gerenciamento de processos

Exemplo

\$ ps aux

```
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
root 1 0.0 0.0 204764 7008 ? Ss Jul17 0:01 /sbin/init
root 2 0.0 0.0 0 0 ? S Jul17 0:00 [kthreadd]
root 3 0.0 0.0 0 0 ? S Jul17 0:00 [ksoftirqd/0]
...
avahi 608 0.0 0.0 47016 360 ? S Jul17 0:00 avahi-daemon: chroot helper
...
mysql 835 0.0 0.9 686288 78416 ? Ssl Jul17 0:17 /usr/sbin/mysqld
...
rafael 1894 0.0 0.0 65132 6620 ? Ss Jul17 0:00 /lib/systemd/systemd -user
rafael 1895 0.0 0.0 232380 1936 ? S Jul17 0:00 (sd-pam)
...
rafael 1986 0.0 0.6 370924 49380 ? Sl Jul17 0:04 xfce4-session
```

Gerenciamento de processos

Exemplo - filtrando o usuário

```
$ ps aux | grep [usuário]
```


Gerenciamento de processos

Exemplo - filtrando o usuário

```
$ ps aux | grep [usuário]
```

```
$ ps aux | grep rafael
```

Gerenciamento de processos

Exemplo - filtrando o usuário

```
$ ps aux | grep [usuário]
```

```
$ ps aux | grep raphael
```

```
rafael 1894 0.0 0.0 65132 6620 ? Ss Jul17 0:00 /lib/systemd/systemd -user
rafael 1895 0.0 0.0 232380 1936 ? S Jul17 0:00 (sd-pam)
rafael 1976 0.0 0.0 11084 332 ? Ss Jul17 0:00 /usr/bin/ssh-agent startxfce4
rafael 1986 0.0 0.6 370924 49380 ? Sl Jul17 0:04 xfce4-session
rafael 2186 0.0 0.2 407904 24056 ? Sl Jul17 0:00 light-locker
rafael 2192 0.0 0.0 187548 4940 ? Sl Jul17 0:00 /usr/lib/dconf/dconf-service
rafael 2226 0.0 0.4 572816 36100 ? Sl Jul17 0:14 nm-applet
```

Gerenciamento de processos

Encerrando processos

- ▶ **CTRL-C**: aborta um processo no terminal
- ▶ **CTRL-Z**: coloca um processo em *background*. O comando **jobs** lista os processos em *background*.
- ▶ **fg <número>**: move um processo em *background* para o *foreground*
- ▶ **kill <PID>**: mata um processo através do seu número de identificação (PID), por padrão o kill envia o sinal 15 (término do processo), mas para forçar o encerramento pode ser utilizado: **\$ kill -9 PID**
- ▶ **kill -i**: pede confirmação da finalização do processo
- ▶ **kill -v**: informa se o sinal foi enviado ou não
- ▶ **killall [opções] [sinal] <nome-processo>**: permite fechar processos pelo nome. Pode-se também utilizar: **\$ pkill [nome]**
- ▶ **killall -u user**: finaliza todos os processos que estão sendo executados

Gerenciamento de processos

O comando top

Mostra os processos de modo dinâmico. As informações são atualizadas a cada certo intervalo de tempo (3 segundos por padrão). É possível ordenar a ordem da lista de processos.

\$ top

```
top - 11:43:56 up 23:49, 1 user, load average: 0.25, 0.58, 0.53
Tasks: 238 total, 1 running, 202 sleeping, 34 stopped, 1 zombie
%Cpu(s): 3.3 us, 1.1 sy, 0.0 ni, 95.6 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 8093488 total, 1519892 free, 3295656 used, 3277940 buff/cache
KiB Swap: 0 total, 0 free, 0 used. 4176228 avail Mem

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
2434 rafael 20 0 2280984 390572 169340 S 8.3 4.8 25:35.85 Web Content
2467 rafael 20 0 2731116 526100 110892 S 2.3 6.5 66:06.06 Web Content
751 root 20 0 569892 176712 137984 S 1.7 2.2 16:05.84 Xorg
2366 rafael 20 0 2899052 463948 185944 S 1.3 5.7 36:07.63 firefox
```

Gerenciamento de processos

O comando top

Após executar o comando top, na janela que se abrir tecele:

- ▶ h: para o menu de ajuda
- ▶ k: para matar um processo
- ▶ i: para ignorar os processos zumbis
- ▶ q: para sair do top (ou CTRL-C)

Algumas opções:

- ▶ -d <intervalo>: intervalo de atualização
- ▶ -o <campo>: ordena a lista de acordo com o <campo>, por exemplo, PID, %MEM, ...

Gerenciamento de processos

1. Abra 5 arquivos novos do gedit
2. Utilize o comando `top`. Teste algumas opções (`-d`, `-o`)
3. Com o `ps ux` visualize o PID de um deles e use o comando `kill`. Finalize um deles utilizando `top`.
4. Use o comando `killall` para matar todos os gedit's abertos
5. Abra uma página do navegador
6. Visualize os processos com o `top` e feche o navegador teclando `k` e PID dele
7. Abra um gedit sem usar o `&`, volte ao terminal e tecle `CRTL+C` para fechá-lo
8. Execute o comando `sleep 10` seguido de `CRTL+Z`. Mova o processo para o *foreground*

Solução dos exercícios

1. `gedit 1 & gedit 2 gedit 3 & gedit 4 & gedit 5 &`
ou abra diretamente pelo menu
2. `top -d 1; top -d 2 -o %CPU; top -o %MEM`
3. `ps ux`, `kill PID`; `top` (encontre o PID) e digite `k PID`
4. `killall gedit`
5. Digite `firefox` no terminal (ou abra diretamente pelo menu)
6. Digite `top` no terminal, procure por Firefox e tecle `k` seguido do PID do Firefox para fechá-lo
7. Digite `gedit` no terminal, e feche com `ctrl c`
8. Digite `sleep 10` seguido de `CRTL+Z`; digite `fg` seguido do número exibido após a entrada do `CRTL+Z`

Baixando dados da internet

O programa wget

- ▶ O programa wget é uma das ferramentas GNU
- ▶ Ele vem instalado por padrão nas distribuições Linux e permite baixar dados da internet via terminal
- ▶ É extremamente flexível e uma ferramenta muito poderosa quando se deseja baixar uma grande quantidade de dados dispersos, como repositórios ftp com vários arquivos em pastas e subpastas, salvando os dados recursivamente, de modo a preservar a estrutura original de diretórios. Também permite recomeçar *downloads* interrompidos ou mesmo baixar dados de sites que requerem login

Baixando dados da internet

Exemplo básico

```
$ wget https://www.gnu.org/software/bash/manual/bash.pdf
```

```
-- 2018-07-22 13:38:15 --
```

```
https://www.gnu.org/software/bash/manual/bash.pdf
```

```
Resolving www.gnu.org (www.gnu.org)... 208.118.235.148,  
2001:4830:134:3::a
```

```
Connecting to www.gnu.org (www.gnu.org)|208.118.235.148|:443...  
connected.
```

```
HTTP request sent, awaiting response... 200 OK
```

```
Length: 733821 (717K) [application/pdf]
```

```
Saving to: bash.pdf
```

```
bash.pdf 59%[=====>
```

```
] 424.00K 449KB
```

Baixando dados da internet

Para continuar um *download* interrompido

```
$ wget -c https://www.gnu.org/software/bash/manual/bash.pdf
```

Especificar nome do arquivo

```
$ wget -O gnu-bash.pdf https://www.gnu.org/software/bash/manual/bash.pdf
```

Especificar o local para salvar

```
$ wget -P pasta/subpasta https://www.gnu.org/software/bash/manual/bash.pdf
```

Baixar múltiplas URLs de uma lista

```
$ wget -i lista-de-URLs.txt
```

Baixando dados da internet

Baixa sequência de arquivos numerados de uma página

```
$ wget http://levee.wustl.edu/seismology/book/chapter3/chap3_fr/3_2_0{1..9}.jpg
```

Baixa um site inteiro e salva *offline*

```
$ wget -m -p -k http://seismo.berkeley.edu/~rallen/eps122/
```

Dribla sites que bloqueiam *downloads* não requisitados por navegador

```
$ wget --referer=http://google.com -U "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:47.0) Gecko/20100101 Firefox/47.0" http://nytimes.com
```

Baixar de páginas que requerem *login*

```
$ wget --user=batman --password=morcego http://seismo.berkeley.edu
```

Baixando dados da internet

O programa curl

Alternativamente, pode-se utilizar o comando **curl** para fazer o download de dados. curl e wget são bem parecidos, porém existem algumas diferenças. As principais delas são:

- ▶ curl é provida pela *libcurl*, uma biblioteca multiplataforma
- ▶ suporta pipes
- ▶ não é recursivo
- ▶ suporta mais protocolos (FTP, FTPS, Gopher, HTTP, HTTPS, SCP, SFTP, TFTP, etc)
- ▶ requer -o ou -O para salvar em um arquivo

Baixando dados da internet

```
$ curl -o gnu-bash.pdf https://www.gnu.org/software/bash/manual/bash.pdf
```

```
$ curl -O https://www.gnu.org/software/bash/manual/bash.pdf
```

Para seguir redirecionamento

```
$ curl -O -L https://dev.mysql.com/get/Downloads/MySQL-8.0/mysql-8.0.11-linux-glibc2.12-i686.tar
```

Concatenação e compactação de arquivos

O formato *tar*

- ▶ O formato *tar* é usado para concatenar múltiplos arquivos em um único
- ▶ Diferentemente de outros formatos mais conhecidos como *zip* ou *rar*, ele apenas concatena, sem comprimir
- ▶ Para comprimir um *tar* utiliza-se outros formatos, como o *gzip*

Sintaxe básica

```
$ tar -c -f concatenado.tar arquivo1 arquivo2 ...
```

Para imprimir na tela o que o comando está fazendo

```
$ tar -c -v -f concatenado.tar arquivo1 arquivo2 ...
```

Uma forma mais compacta de escrever

```
$ tar -cvf concatenado.tar arquivo1 arquivo2 ...
```

Concatenação e compactação de arquivos

Significado das opções

- c: concatenar
- v: *verbose*, ou seja, imprime na tela mais informações
- f: nome do arquivo gerado

Concatenar e comprimir de uma só vez utilizando gzip

```
$ tar -cvzf concatenado.tar.gz arquivo*
```

Descomprimir e desconcatenar de uma só vez

```
$ tar -xvzf concatenado.tar.gz
```

Outras opções de compressão

- ▶ j: bzip2
- ▶ J: xz

Concatenação e compactação de arquivos

Formato zip

- ▶ `zip comprimido.zip arquivo1 arquivo2 ...`
- ▶ `unzip comprimido.zip`

Exercícios

1. Usando o `wget` faça o download da aula
`http://seismo.berkeley.edu/~rallen/eps122/lectures/L01.pdf`
e salve com o nome `aula1.pdf`
2. Do mesmo repositório salve as aulas L01 até L05
3. Baixe as aulas usando os links do arquivo `aulas.txt` e salve numa nova pasta chamada *aulas* dentro da pasta *exercícios*
4. Concatene e comprima os arquivos (dentro da pasta `arquivos`)
nomeados de `A*` no formato `tar.gz`. Teste as opções de compactação
e compare
5. Descomprima e desconcatene o arquivo criado
6. Comprima os mesmos arquivos no formato `zip`

SSH - Secure Shell



SSH - Secure Shell

O SSH é um protocolo de rede criptografado desenvolvido em sua primeira versão em 1995 por Tatu Ylönen, na época pesquisador da Universidade de Helsinki na Finlândia. Esse protocolo foi desenvolvido para substituir o rlogin, TELNET e rsh que não traziam muita confiabilidade para os usuários por não fornecer uma autenticação muito segura.

A função do SSH é fornecer um canal de conexão segura entre o usuário e o servidor, tendo como uma das atividades mais comuns o acesso a computadores remotos. Para isso usa criptografia de chave pública para a autenticação do servidor e do usuário.

SSH - Secure Shell

O SSH é um protocolo de rede criptografado desenvolvido em sua primeira versão em 1995 por Tatu Ylönen, na época pesquisador da Universidade de Helsinki na Finlândia. Esse protocolo foi desenvolvido para substituir o rlogin, TELNET e rsh que não traziam muita confiabilidade para os usuários por não fornecer uma autenticação muito segura.

A função do SSH é fornecer um canal de conexão segura entre o usuário e o servidor, tendo como uma das atividades mais comuns o acesso a computadores remotos. Para isso usa criptografia de chave pública para a autenticação do servidor e do usuário.

OpenSSH - Open Secure Shell

O OpenSSH apareceu em 1999 desenvolvido por Aaron Campbell, Bob Beck, Markus Friedl, Niels Provos, Theo de Raadt e Dug Song. É baseado na última versão livre do SSH com algumas correções de falhas de segurança e outras melhorias.

Acessando outro computador pelo SSH

`ssh [opções] usuario@servidor`

- ▶ **ssh servidor:** Só se aplica caso o usuário do servidor seja o mesmo do atual
- ▶ **ssh servidor -l user ou ssh user@servidor:** Loga no servidor com a conta user
- ▶ **ssh -C user@server:** Conecta ao servidor usando compactação
- ▶ **ssh -X user@server:** Habilita a execução de aplicativos gráficos (X11 forwarding), mas considera o servidor como não confiável, caso o comando viole alguma configuração de segurança aparecerá uma mensagem de erro e o aplicativo não será executado
- ▶ **ssh -Y user@server:** Mesma função do -X contudo considera o servidor confiável e executa o aplicativo mesmo que viole algum critério de segurança. Essa opção só deve ser usada em servidores confiáveis

Transferência de dados

- ▶ Transferir um arquivo do computador para o servidor
`scp ~/Documents/cursoLinux/Aula1/aula1.pdf sismo50@tropic:~/cursoLinux`
- ▶ Copiar um arquivo do servidor para o computador
`scp sismo50@tropic:~/cursoLinux/aula1.pdf ~/Documents/cursoLinux/Aula1`
- ▶ Enviar um diretório e todo seu conteúdo para o servidor
`scp -r ~/Documents/cursoLinux/Aula1 sismo50@tropic:~/cursoLinux`
- ▶ Enviar um diretório e todo seu conteúdo para o servidor e preservar os dados de modificação e acesso do arquivo original
`scp -rp ~/Documents/cursoLinux/Aula1 sismo50@tropic:~/cursoLinux`
- ▶ Habilitar a compressão do arquivo `scp -C`

Transferência de dados - rsync

```
$ rsync -e ssh arquivo user@servidor:/caminho/
```

Transferência de dados - rsync

Algumas opções:

-a: modo *arquivo*. Inclui:

-r: recursão

-l: copia links simbólicos como links simbólicos

-p: preserva permissões

-t: preserva a data de modificações

-g: preserva o grupo

-o: preserva o usuário (apenas no modo super-usuário)

-D: preserva arquivos de dispositivo (por exemplo, drivers) e arquivos especiais (ambos apenas no modo super-usuário)

-z: comprime o arquivo durante a transferência

-h: utiliza números em formatos “legíveis” para humanos

-v: aumenta as informações de saída do programa

--progress: mostra o progresso durante a transferência

Acesso remoto

Transferência de dados - rsync

Copia um arquivo específico

```
$ rsync -avzh -e ssh ~/Documents/cursoLinux/Aula1/aula1.pdf sismo50@tropic:~/cursoLinux/
```

Copia os arquivos da pasta Aula1:

```
$ rsync -avzh -e ssh ~/Documents/cursoLinux/Aula1/ sismo50@tropic:~/cursoLinux/
```

Copia a pasta Aula1:

```
$ rsync -avzh -e ssh ~/Documents/cursoLinux/Aula1 sismo50@tropic:~/cursoLinux/
```

Copia do servidor para a máquina local:

```
$ rsync -avzh -e ssh sismo50@tropic:~/cursoLinux/aula1.pdf  
~/Documents/cursoLinux/aula1-do-servidor.pdf
```

Exercícios

1. Acesse o servidor tropic2 utilizando o usuário sismo(número da sua máquina)
2. Crie uma pasta com seu nome e depois de um exit para voltar à máquina virtual
3. Copie a pasta que você criou na tropic2 para sua home
4. Copie o arquivo temaki.dat da pasta aula3 para sua pasta na tropic2
5. Repita o comando com e sem a opção -p salvando o arquivo com nome diferente, e compare a diferença na data de modificação do arquivo
6. Copie o arquivo temaki.dat utilizando o rsync.

Solução dos Exercícios

1. `$ ssh sismo50@tropic2`
2. `$ mkdir Nome; exit`
3. `$ scp -r sismo50@tropic2:~/Nome ~`
4. `$ scp ~/aula3/temaki.dat sismo50@tropic2:~/Nome`
5. `$ scp -p ~/aula3/temaki.dat sismo50@tropic2:~/Nome/temaki2.dat`
6. `$ rsync -avzh -e ssh ~/aula3/temaki.dat sismo50@tropic2:~/Nome/`

convert

O programa **convert** é um membro da suíte de ferramentas ImageMagick (<https://www.imagemagick.org/script/convert.php>).

Esse programa pode ser usado para converter os formatos de imagens, reescalar uma imagem, girar, juntar, entre muitas outras opções.

Alterar o formato

```
$ convert lenna.png lenna.jpg
```

Alterar o tamanho

```
$ convert lenna.png -resize 50% lenna-50.png
```

Exercícios

1. Altere o tamanho da imagem utilizando uma medida em pixels, por exemplo, 50x80. Teste diferentes valores.
2. Repita os testes mas acrescente “!” após os valores das medidas.
3. Altere a qualidade utilizando a opção “-quality <valor>”. Teste diferentes valores.

Solução dos Exercícios

1. `$ convert lenna.png -resize 100x100 lenna-100.png`
2. `$ convert lenna.png -resize 200x300! lenna-200.png`
3. `$ convert lenna.png -quality 90 lenna-q90.png`

Editores de Texto

Existem diversos de editores de texto desenvolvidos para Linux. Alguns abrem o texto no próprio terminal e outros possuem interface gráfica e opções mais avançadas. A escolha do editor com ou sem interface costuma vir do custo-benefício. Por exemplo, se é preciso inserir apenas um pequeno comentário no texto ou deletar uma linha, os editores de linha de comando atendem bem, porém se queremos escrever um código mais extenso ou um texto bem estruturado vale a pena um editor gráfico.

Editores de linha de comando

- ▶ Vi ou Vim
- ▶ Nano
- ▶ Joe

Editores com interface gráfica

- ▶ Gedit, mousepad
- ▶ LibreOffice Writer
- ▶ L^AT_EX

Editores de linha de comando (CLI)

Vi ou Vim

```
$ vim arquivo.txt  
$ vim
```

O vim possui o **modo de inserção** para adicionar texto e o **modo de comando** que é usado, por exemplo, para excluir palavras, linhas, salvar e fechar o arquivo.

Modo de comando

- ▶ Mover o cursor para cima: seta para cima ou k
- ▶ Mover o cursor para baixo: seta para baixo ou j
- ▶ Mover o cursor para direita: seta para direita ou l
- ▶ Mover o cursor para esquerda: seta para esquerda ou h

Modo de comando

- ▶ **dl**: Exclui a primeira letra a frente do cursor
- ▶ **dw**: Exclui a primeira palavra a frente do cursor
- ▶ **d3w**: Exclui as três primeiras palavras frente do cursor
- ▶ **dd**: Exclui a linha toda
- ▶ **d\$**: Exclui toda a linha a frente do cursor
- ▶ **yy ou Y**: Copia toda a linha
- ▶ **p**: Cola o que está na memória após a linha do cursor
- ▶ **P**: Cola o que está na memória antes da linha do cursor
- ▶ **rA**: Troca o caractere a frente do cursor por A

Vi ou Vim

Modo de comando

Vi ou Vim

Modo de comando

- ▶ **:** → Ativa alinha de comando na parte inferior
- ▶ **:q** → Sai sem salvar, o programa irá te avisar caso haja alguma alteração não salva
- ▶ **:q!** → Sai sem salvar e nenhuma mensagem é exibida
- ▶ **:w** → Salva o arquivo
- ▶ **:w nome** → Salva o arquivo com o nome dado, caso exista outro arquivo com esse nome será dado um aviso, para sobrescrever use **:w! nome**
- ▶ **:wq** → Salva o arquivo e sai do vim

Modo de inserção

Tecla **i** no modo de comando para ativar a inserção de texto, use as setas do teclado pra navegar e o delete e backspace para apagar caracteres inseridos. Para voltar ao modo comando basta teclar **ESC**.

Exercícios

1. Abra o arquivo `pao_de_queijo.dat` com o vim e use o modo comando para as atividades a seguir:
2. Remova a palavra Ingredientes duplicada;
3. Remova o queijo parmesão ralado em excesso;
4. Troque o m de modo de preparo para M;
5. Navegue pela tela usando as teclas h j k e l;
6. Ative o modo de inserção e acrescente uma linha com rendimento da receita;
7. Volte ao modo de comando, salve e saia do vim.

Solução dos Exercícios

1. `vim pao_de_queijo.dat`
2. Mova o cursor a frente da palavra e digite **`dw`**
3. Mova o cursor a frente da palavra queijo e digite **`d3w`**
4. Mova o cursor a frente da letra **`m`** e tecele **`rM`**
5. Prático
6. tecele **`i`** para ativar o modo de inserção
7. tecele **`esc`** para voltar ao modo de comando e tecele **`:wq`** para salvar.

Editores com interface gráfica



Editores com interface gráfica

Gedit

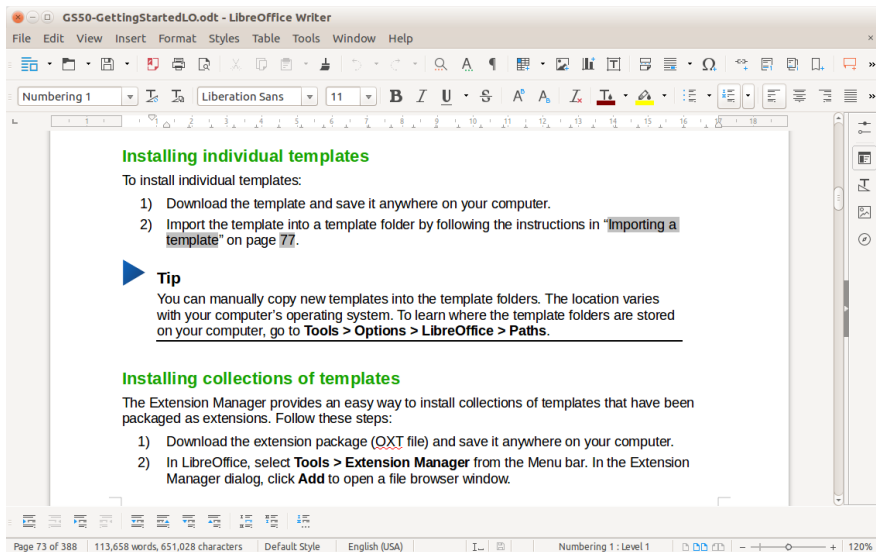
O gedit provavelmente é o editor mais usado em Linux e possui muitas funcionalidades desde que sejam adicionados alguns plugins. Esses plugins trazem uma série de vantagens e facilidades para se trabalhar com alguma linguagem de programação. Para acessar a tela com as opções de plugins clique em editar → preferências → plugins. Vejamos alguns:

- ▶ Fechar automaticamente colchetes, chaves e parênteses
- ▶ Habilitar comentário do código teclando CTRL+M e descomentar com CTRL+SHIFT+M
- ▶ Desenhar espaços e tabulações
- ▶ Corretor ortográfico
- ▶ Completar palavras
- ▶ Mudar a cor de fundo da linha onde está o cursor

LibreOffice



LibreOffice - Writer



The screenshot shows a LaTeX editor window with the following components:

- Document Title:** Document : /home/joseph/Documents/tex/listingsandminted/foo-bar.tex
- Structure Panel:**
 - foo-bar.tex
 - LABELS
 - Hello World
 - Hi The...
- Source Code Editor:**

```

8 \section{Hi There, John}
9 \emph{Baz}
10 \begin{tabular}{ccc}
11   Foo & Bar & Baz \\ \hline
12   1   & 2   & 3   \\
13   4   & 5   & 6   \\
14 \end{tabular}
15 and then a list with cross
16   references and citations
17 \begin{enumerate}
18   \item \label{it:first} This is
19     an item with a citation
20     \cite{dummy}.
21   \item This item has a reference
22     to the previous item, namely
23     \ref{it:first} and should be
24     deliberately longer than 80
25     characters.
26 \end{enumerate}
27 \end{document}

```
- Rendered Output:**

Inline math mode: $\int_0^{2\pi} \sin x \, dx$. Display math mode:

$$\int_0^{2\pi} \sin x \, dx$$

Something verbatim here with \$

1.1 Hi There, John

| Baz | Foo | Bar | Baz |
|-----|-----|-----|-----|
| 1 | 2 | 3 | |
| 4 | 5 | 6 | |

and then a list with cross references

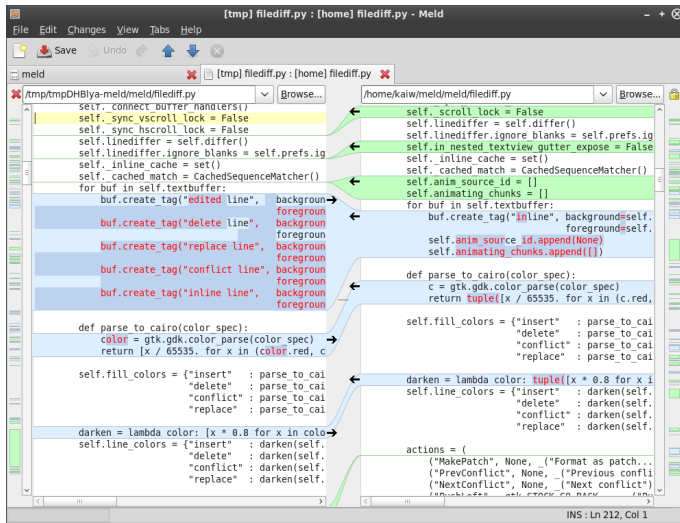
 1. This is an item with a citation [?].
 2. This item has a reference to the previous item, namely deliberately longer than 80 characters.
- Status Bar:** Line : 18 Visual column : 72 Text column : 72 INSERT
- Messages Panel:**

```

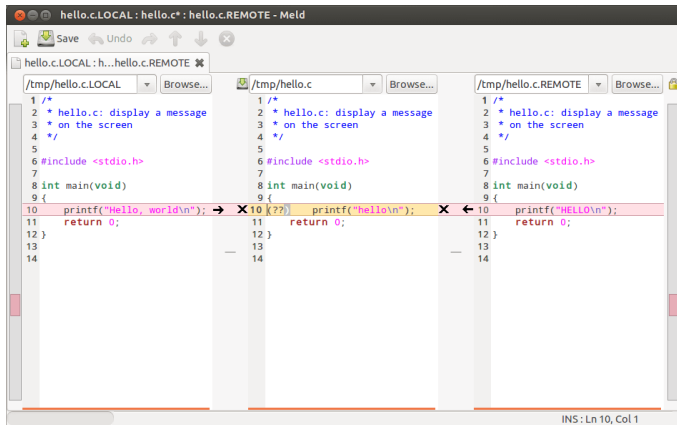
ts: type 1/public/amsfonts/cm/cmt10.pro>
Output written on foo-bar.pdf (1 page, 85956 bytes).
SyncTeX written on foo-bar.synctex.gz.
Transcript written on foo-bar.log.

Process exited normally

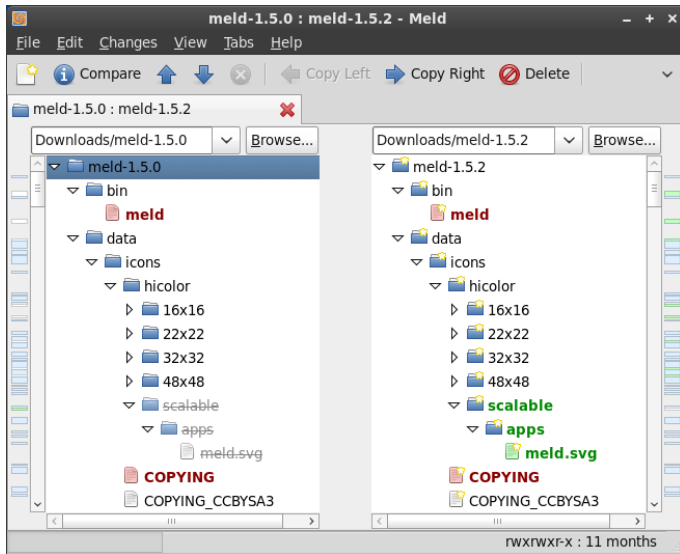
```



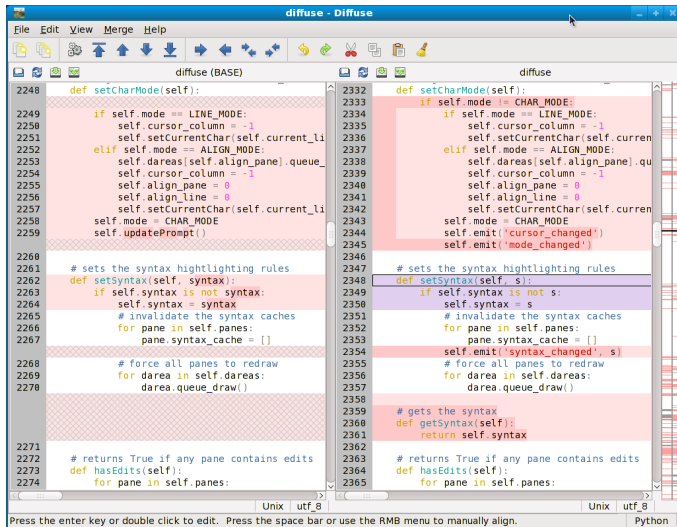
Meld



Meld



Diffuse



```
diffuse - Diffuse
File Edit View Merge Help
diffuse (BASE)
2248 def setCharMode(self):
2249     if self.mode == LINE_MODE:
2250         self.cursor_column = -1
2251         self.setCurrentChar(self.current_li
2252     elif self.mode == ALIGN_MODE:
2253         self.dareas[self.align_pane].queue_
2254         self.cursor_column = -1
2255         self.align_pane = 0
2256         self.align_line = 0
2257         self.setCurrentChar(self.current_li
2258         self.mode = CHAR_MODE
2259         self.updatePrompt()
2260
2261 # sets the syntax highlighting rules
2262 def setSyntax(self, syntax):
2263     if self.syntax is not syntax:
2264         self.syntax = syntax
2265         # invalidate the syntax caches
2266         for pane in self.panes:
2267             pane.syntax_cache = []
2268
2269 # force all panes to redraw
2270 for darea in self.dareas:
2271     darea.queue_draw()
2272
2273 # returns True if any pane contains edits
2274 def hasEdits(self):
2275     for pane in self.panes:
2276
diffuse
2332 def setCharMode(self):
2333     if self.mode != CHAR_MODE:
2334         if self.mode == LINE_MODE:
2335             self.cursor_column = -1
2336             self.setCurrentChar(self.curren
2337         elif self.mode == ALIGN_MODE:
2338             self.dareas[self.align_pane].qu
2339             self.cursor_column = -1
2340             self.align_pane = 0
2341             self.align_line = 0
2342             self.setCurrentChar(self.curren
2343             self.mode = CHAR_MODE
2344             self.emit('cursor_changed')
2345             self.emit('mode_changed')
2346
2347 # sets the syntax highlighting rules
2348 def setSyntax(self, s):
2349     if self.syntax is not s:
2350         self.syntax = s
2351         # invalidate the syntax caches
2352         for pane in self.panes:
2353             pane.syntax_cache = []
2354             self.emit('syntax_changed', s)
2355         # force all panes to redraw
2356         for darea in self.dareas:
2357             darea.queue_draw()
2358
2359 # gets the syntax
2360 def getSyntax(self):
2361     return self.syntax
2362
2363 # returns True if any pane contains edits
2364 def hasEdits(self):
2365     for pane in self.panes:
2366
Unix utf_8 Python
Press the enter key or double click to edit. Press the space bar or use the RMB menu to manually align.
```

Leitores de PDF

gv - ghost viewer

Leitor antigo, com poucas funcionalidades mas bem leve e agil. Para pdf já não é mais a primeira escolha mas funciona muito bem com imagens ps (post script)

Evince

Interface simples e sem muitas opções mas da conta das necessidades básicas, costuma ser o leitor padrão de pdf

Okular

O mais sofisticado dos três e o que mais se assemelha ao Adobe reader, possui uma série de opções para adição de comentários (revisão)

Exercícios

1. Abra a aula de hoje com o evince, com o okular e com o gv (não se esqueça de acrescentar o & no final do comando para não travar o terminal)
2. Abra um novo arquivo de edição de texto do libreoffice
3. Abra uma nova planilha
4. Habilite os plugins do gedit de completar palavras desenhar espaços em branco

EXTRAS



Extra - Nano

- ▶ ctrl g (^g) ou F1 : Exibe a tela e ajuda;
- ▶ ctrl x ou F2: Fecha a tela atual/ sai do nano;
- ▶ ctrl o ou F3: Salva o arquivo;
- ▶ ctrl j ou F4: Justifica o paragrafo onde está o cursor;
- ▶ ctrl r ou F5: Insere um arquivo, ou abre um arquivo;
- ▶ ctrl w ou F6: Procura por uma palavra ou expressão;
- ▶ ctrl y ou F7: Vai para a tela anterior;
- ▶ ctrl v ou F8: Vai para a tela posterior;
- ▶ ctrl k ou F9: Corta a linha;
- ▶ ctrl u ou F10: Cola o conteúdo armazenado;
- ▶ ctrl c ou F11: Exibe a posição do cursor;
- ▶ ctrl t ou F12: Ativa o verificador ortográfico se disponível;
- ▶ alt a: Ativa a seleção, basta mover o cursor para selecionar.

Exercícios

1. Abra um arquivo em branco com o nano;
2. Leia o arquivo `pao_de_queijo.dat`;
3. Carregue o arquivo `temaki` no final do atual;
4. Use as ferramentas de cortar e colar para ordenar ingredientes e modo de preparo das duas receitas;
5. Visualize em que linha do arquivo e posição na coluna começa o primeiro modo de preparo;
6. Pesquise pela palavra `água` no arquivo, use `alt w` para navegar por todas as aparições;
7. Salve o arquivo com um novo nome.

Solução dos Exercícios

1. `vim a`
2. `vim pao_de_queijo.dat`
3. `ctrl R temaki.dat`
4. `ctrl k` - cortar, `ctrl u` - colar
5. `ctrl c;`
6. `ctrl w`
7. `ctrl o novonome`

Extra - joe

- ▶ ctrl kh: Tela de ajuda;
- ▶ ctrl kf: Pesquisa um termo ou expressão;
- ▶ ctrl l: Navega pelos itens encontrados na pesquisa;
- ▶ ctrl ku: Move o cursor para a primeira linha;
- ▶ ctrl kv: Move o cursor para a última linha;
- ▶ ctrl kb: Marca o começo da seleção;
- ▶ ctrl kk: Marca o final da seleção;
- ▶ ctrl kb: Cola o trecho selecionado onde está o cursor;
- ▶ ctrl w: Deleta a palavra a frente do cursor;
- ▶ ctrl y : Deleta a linha onde está o cursor;
- ▶ ctrl kx: Salva e fecha o joe;
- ▶ ctrl c: Aborta a execução do joe.

Exercícios

1. Abra arquivo bolo_cenoura.dat na pasta de exercícios da aula;
2. Pesquise por xícara e navegue para as próximas ocorrências;
3. Copie o modo de preparo da massa no fim do arquivo;
4. Use os atalhos de teclado para ir com o cursor para a primeira e última linha;
5. Delete o bloco copiado no fim do arquivo linha por linha;
6. Salve e feche o arquivo;

Solução dos Exercícios

1. `joe bolo_cenoura.dat`
2. `ctrl f` pesquisa, `ctrl l` navega
3. `ctrl kb` marca o início da seleção, `ctrl kk` o final e `ctrl kb` cola o trecho selecionado onde está o cursor
4. `ctrl ku` - primeira linha e `ctrl kv` - última linha
5. `ctrl y`
6. `ctrl kx`